

# Sécurité du service de courrier électronique : amavisd-new

Philippe Latu

philippe.latu(at)linux-france.org

<http://www.linux-france.org/prj/inetdoc/>

Historique des versions		
\$Revision: 722 \$	\$Date: 2005-11-10 16:53:34 +0100 (jeu, 10 nov 2005) \$	PL
Draft en cours de rédaction		

## Table des matières

1. Copyright et Licence .....	2
1.1. Meta-information .....	2
2. Le contexte sécurité du service de courrier électronique .....	2
2.1. L'ingénierie sociale ou les limites de la technologie .....	2
2.2. Le placement des fonctions de sécurisation du courrier électronique .....	3
2.3. L'architecture système .....	5
2.4. L'architecture réseau .....	6
2.5. L'architecture «domestique» .....	7
2.6. Les règles de filtrage réseau .....	7
3. Le service amavisd-new .....	8
3.1. L'installation .....	8
3.1.1. Le téléchargement des sources et les dépendances .....	8
4. Les antivirus .....	9
4.1. Sophos .....	10
4.1.1. SAV interface .....	10
4.1.2. Sophie .....	10
5. Les échantillons relevés .....	11
5.1. Module MIME::Tools .....	11
6. Documents de référence .....	11

# 1. Copyright et Licence

Copyright (c) 2000,2005 Philippe Latu.  
 Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2005 Philippe Latu.  
 Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.1 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

## 1.1. Meta-information

Cet article est écrit avec *DocBook*<sup>1</sup> XML sur un système *Debian GNU/Linux*<sup>2</sup>. Il est disponible en version imprimable aux formats PDF et Postscript : [amavisd-new.pdf](#)<sup>3</sup> | [amavisd-new.ps.gz](#)<sup>4</sup>.

# 2. Le contexte sécurité du service de courrier électronique

«La sécurité est un processus et non un produit. Et comme dans tout processus, certaines de ces composantes sont plus solides, plus fiables, mieux huilées et plus sûres que d'autres. De plus, ces composantes doivent s'emboîter les unes dans les autres. Mieux elles s'emboîtent, mieux le processus fonctionne. Souvent, ce sont les interfaces entre les composantes qui sont les éléments les moins sûrs.»

--Secrets et mensonges - Sécurité numérique dans un monde en réseau - Bruce Schneier

## 2.1. L'ingénierie sociale ou les limites de la technologie

L'humain est-il la composante la plus faible du processus de sécurisation du courrier électronique ? Répondre oui à cette question c'est choisir un alibi facile. L'ingénierie sociale (ou «esbrouffe» dans un français plus académique), montre simplement les limites de la technologie. Il n'existe pas (encore ?) de système technique capable de détecter toutes les formes d'usurpation d'identité.

- Ouvrir la pièce jointe encryptée (contenant le virus !) dont la clé de déchiffrement est fournie dans le corps du message est un piège facile à éviter tant que l'on est pas «sensible» au contenu dudit message. Seule l'interprétation humaine provoquera le déclenchement du code viral.
- Compléter un formulaire demandant toutes les coordonnées bancaires (n° de carte de crédit, etc.) transmis par courrier est un piège facile à éviter tant que l'on «détecte une différence» entre le message et le formulaire Web que l'on a l'habitude de saisir pour consulter l'état de son compte en banque. Là encore, tout n'est qu'une question d'interprétation qui échappe totalement aux technologies de sécurité.

Face aux évolutions constantes des méthodes d'usurpation d'identité, seule une information constante et complète des utilisateurs sur le «bon usage» du service de courrier électronique est efficace. Cette sensibilisation sur les usages, aussi nécessaire soit-elle, est parfois difficile. Essayez de convaincre la totalité des utilisateurs d'un service de ne plus composer leurs courriers en HTML !. Gros challenge en perspective ;).

Même si la technologie n'est pas une arme absolue, ses capacités de traitements automatisés sur des volumes de

<sup>1</sup> <http://www.docbook.org>

<sup>2</sup> <http://www.debian.org>

<sup>3</sup> <http://www.linux-france.org/prj/inetdoc/telechargement/amavisd-new.pdf>

<sup>4</sup> <http://www.linux-france.org/prj/inetdoc/telechargement/amavisd-new.ps.gz>

messages importants rendent de grands services.

L'objet de ce document est justement de présenter un service *interface* entre les fonctions classiques de filtrage de contenus de courrier électronique.

## 2.2. Le placement des fonctions de sécurisation du courrier électronique

L'acheminement du courrier passe par des (étapes|points) bien particuliers. Tout commence par le service de noms de domaines (DNS) qui désigne les adresses IP responsables du traitement du courrier électronique d'une zone donnée. Les hôtes responsables de ces traitements (*Mail Transfer Agent*) sont repérés par le champ *Mail eXchanger* (MX) dans le fichier de zone DNS.

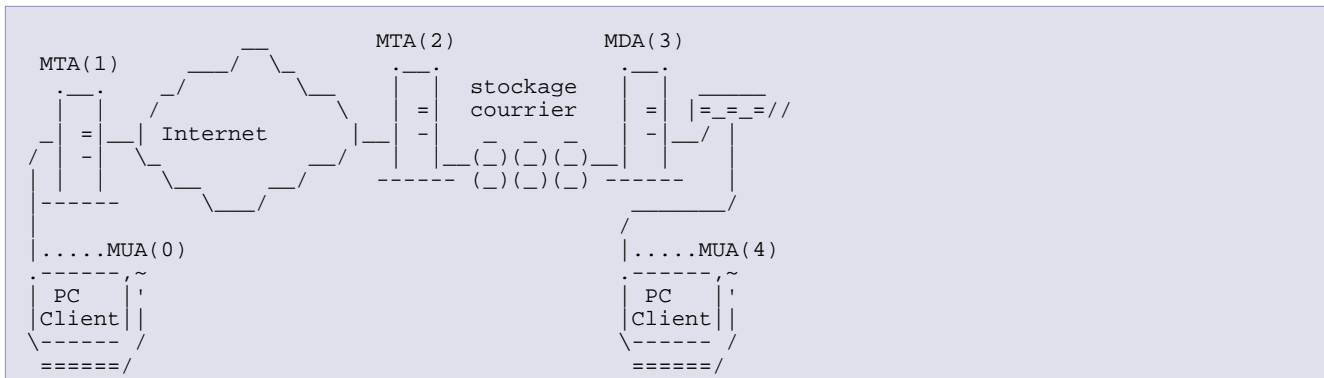
Voici un exemple simple permettant d'obtenir la liste des hôtes responsables du courrier électronique pour la zone linux-france.org. :

```
$ dig nic.fr MX
; <<>> DiG 9.3.1 <<>> nic.fr MX
;; global options: printcmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 1648
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 7, ADDITIONAL: 3

;; QUESTION SECTION:
;nic.fr.                IN      MX

;; ANSWER SECTION:
nic.fr.                172800 IN     MX     50 mx1.nic.fr.
nic.fr.                172800 IN     MX     100 mx2.nic.fr.
<snipped/>
```

Passons maintenant au courrier électronique proprement dit et son acheminement entre zones de l'Internet.



### MUA(0), Mail User Agent, Emission du courrier

L'utilisateur émet un courrier à l'aide d'une application appelée *Mail User Agent* vers le **MTA(1)** de son fournisseur d'accès Internet. La seule précaution possible à ce niveau consiste à utiliser un antivirus «à jour». Concrètement, cette condition est rarement respectée sur les postes domestiques. Pire encore, la vitesse de propagation des vers est maintenant supérieure à la fréquence des mises à jour de signatures de virus. Le niveau de sécurité d'un poste émettant du courrier est donc *nécessairement* faible.



### Avertissement

Il est possible, via une configuration particulière, d'émettre directement du courrier vers n'importe quel MTA sur l'Internet à partir du poste client.

Ce type de fonctionnement correspond presque systématiquement à une infection virale. C'est pour cette raison que dès que ces émissions sont détectées, l'adresse IP du poste est classée en liste noire. Il est donc vivement déconseillé d'émettre directement du courrier vers d'autres MTA que celui de son fournisseur d'accès.

### MTA(1), Mail Transfer Agent (-TX), Transfert du courrier en émission

Le rôle du *Mail Transfer Agent* est de transférer le courrier sur l'Internet vers le *Mail Transfer Agent*

correspondant à l'adresse du destinataire.

La sécurisation des communications entre les MTAs est à la charge des opérateurs qui acheminent les données. Le rôle des fournisseurs d'accès dans la propagation des vers n'est pas neutre. Les clients sont facturés deux fois pour un service qui leur est dû. La surconsommation de bande passante due aux «pourriels» et aux vers est facturée à travers les forfaits et les fonctions antivirus et antispam sont facturées à travers des prestations supplémentaires. Si le problème était traité à la source, la surconsommation de bande passante serait sensiblement diminuée et la propagation des vers limitée.

Outre la logique commerciale de la démarche, l'argumentaire sur l'approche globale de la sécurisation s'applique parfaitement ici. Tant qu'un opérateur «respecte» la segmentation du marché qui lui impose d'affecter un groupe d'unités centrales à une fonction unique (antivirus, antispam, listes noires, etc.) sans (*organiser/contrôler*) les relations entre ces fonctions, il n'a d'autre choix que d'investir sans fin dans de nouveaux châssis dédiés. Cette course est asphyxiante aussi bien sur le plan financier que sur le plan des ressources humaines disponibles pour administrer le service.

Dans ce contexte, il faut *réagir* à son niveau et prendre un maximum de précautions lors de l'émission de courrier électronique depuis sa propre infrastructure. On observe trop souvent des comportements *irresponsables* d'administrateurs, qui sous prétexte que leur installation n'a pas été infectée, n'appliquent aucun traitement sur les courriers qui transitent par leur service alors qu'ils ne leur sont pas destinés.

Les fonctions de sécurité sont identiques à celles mises en oeuvre au niveau d'un MTA(2) de réception du courrier électronique. Seules les relations entre ces fonctions diffèrent. C'est au service de réception d'assumer la protection des périmètres placés sous son contrôle.

#### MTA(2), *Mail Transfer Agent* (-RX), Transfert du courrier en réception

Relativement au MTA(1), le *Mail Transfer Agent* de réception joue le rôle le plus important au niveau sécurité. C'est à ce point de passage que la charge de sécurisation est la plus critique. Il n'est pas rare de devoir «jeter» plus de 30% des courriers présentés au *Mail Transfer Agent*. De plus, le début d'année 2004 a montré combien il est important que les relations entre les fonctions de sécurité soient bien contrôlées. Cette maîtrise de la chaîne de sécurisation permet «d'encaisser» les chocs lors des propagations de nouveaux vers.

#### MDA(3), *Mail Delivery Agent*, Délivrance du courrier

Le rôle du *Mail Delivery Agent* est de prélever le courrier dans les files d'attentes et de le déposer dans le répertoire de boîte aux lettres de l'utilisateur. Procmail est l'outil MDA le plus utilisé dans l'univers GNU/Linux. Il est possible de placer des fonctions de sécurité à ce niveau : appels antivirus (et/ou) antispam. Ce type d'appel est très pénalisant en charge CPU et surtout en accès au système de fichiers sur la machine qui exécute le programme. Il est donc déconseillé de traiter de gros volumes de courrier de cette façon.

Malgré ce défaut très pénalisant, le *Mail Delivery Agent* est l'outil de personnalisation des fonctions de sécurité. Si l'utilisateur souhaite régler lui-même les paramètres de fonctionnement des outils de sécurité, c'est là que l'opération doit se faire. Ces réglages individuels ne sont pas incompatibles avec les fonctions mises en oeuvre au niveau *Mail Transfer Agent*.

#### MUA(4), *Mail User Agent*, Réception du courrier

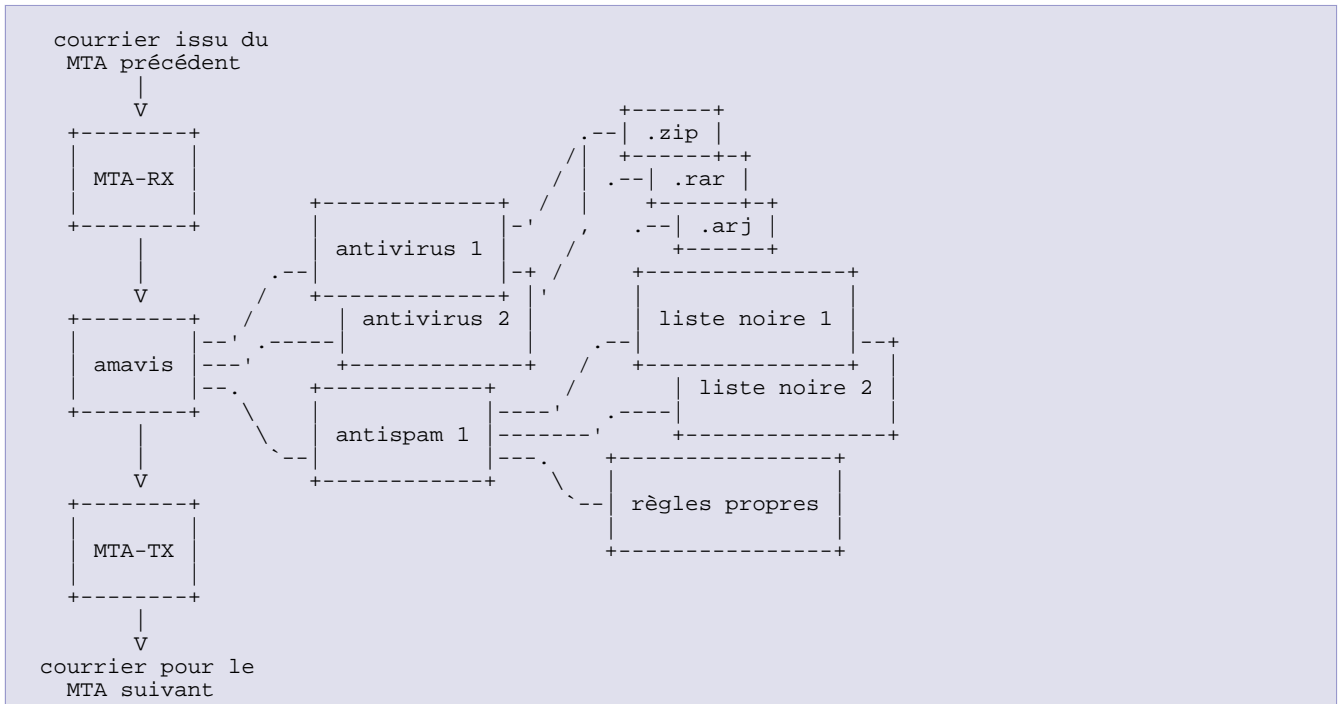
Pour faire simple, si un courrier infecté arrive à ce niveau : c'est foutu ! La population des administrateurs de parc informatique ayant constaté que les antivirus et antispam clients sont totalement inutiles croît à une vitesse exponentielle. Si tous les ténors des solutions propriétaires cherchent encore à faire illusion, on (entend)lit de plus en plus que la vitesse de propagation des vers et des pourriels est beaucoup trop rapide pour que ces solutions propriétaires soutiennent le rythme.

Dans la suite de ce document, nous allons nous concentrer sur les fonctions de sécurité appliquées aux *Mail Transfer Agents* assurant l'émission (-TX) et la réception (-RX) du courrier à la frontière du périmètre administré. Comme indiqué ci-avant (MDA(3)), il est toujours possible d'affiner les paramètres d'utilisation des outils de sécurité. C'est le moyen de personnaliser au niveau utilisateur la chaîne de sécurité. Il faut rappeler que cette individualisation a un coût d'exécution important. Par exemple, les appels à spamassassin à travers procmail sont si longs à exécuter qu'ils gênent la consultation des courriers électroniques.

## 2.3. L'architecture système

En considérant les points de passage obligatoires de l'acheminement du courrier électronique on retient que les *Mail Transfer Agents* situés à la frontière du réseau sont les outils critiques sur lesquels l'approche globale de sécurité du service de courrier électronique doit s'appliquer.

Pour appliquer cette approche globale on utilise un service dédié aux relations entre les fonctions de sécurité : **amavisd-new**.



Dans l'exemple ci-dessus, on a représenté 3 fonctions de sécurité à titre indicatif : 2 antivirus et 1 antisпам. Il existe de très nombreuses combinaisons possibles (FIXME: Voir liste amavis). Chaque fonction de sécurité peut elle même faire appel à plusieurs autres services : bases de données de signatures de virus, bases de données de listes noires, algorithmes de calcul de notes, algorithmes de (dé)compression, etc. Donner une représentation exhaustive de toutes les relations possibles serait illisible.

Les en-têtes de courrier servent de plus en plus à (véhiculer|échanger) des paramètres entre les services exécutés par les différents *Mail Transfer Agents*. Ce sont ces échanges de paramètres (`X-Virus-*`, `X-Spam-*`, etc.) qui régissent les relations entre MTA. Le service **amavisd-new** doit donc nécessairement prendre en compte ces champs d'en-têtes pour prétendre à l'approche globale de sécurité.

Voici un «bel exemple» d'utilisation des champs d'en-tête indiquant le résultat d'un calcul de pondération antisпам :

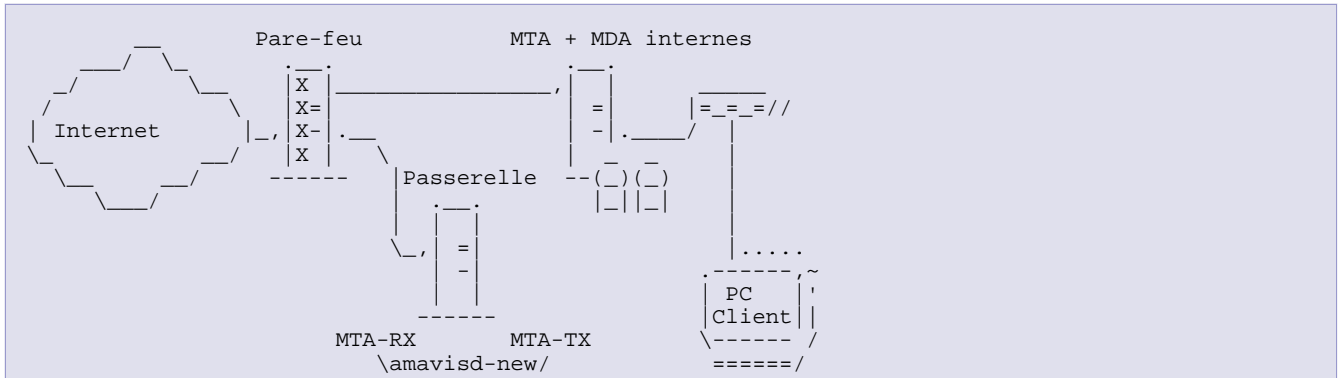
```

X-Spam-Status: Yes, hits=33.2 tag1=3.0 tag2=8.0 kill=8.0
tests=CLICK_BELOW_CAPS, FORGED_YAHOO_RCVD, HTML_FONTCOLOR_RED,
HTML_FONTCOLOR_UNKNOWN, HTML_FONT_BIG, HTML_MESSAGE, HTML_SHOUTING4,
MIME_HTML_ONLY, MIME_HTML_ONLY_MULTI, OBFUSCATING_COMMENT,
RAZOR2_CF_RANGE_51_100, RAZOR2_CHECK, RCVD_IN_BL_SPAMCOP_NET, RCVD_IN_DSBL,
RCVD_IN_DYNABLOCK, RCVD_IN_NJABL, RCVD_IN_NJABL_PROXY, RCVD_IN_OPM,
RCVD_IN_OPM_HTTP, RCVD_IN_OPM_HTTP_POST, RCVD_IN_SBL, RCVD_IN_SORBS,
RCVD_IN_SORBS_HTTP, REMOVE_PAGE, UPPERCASE_25_50
X-Spam-Level: *****
  
```

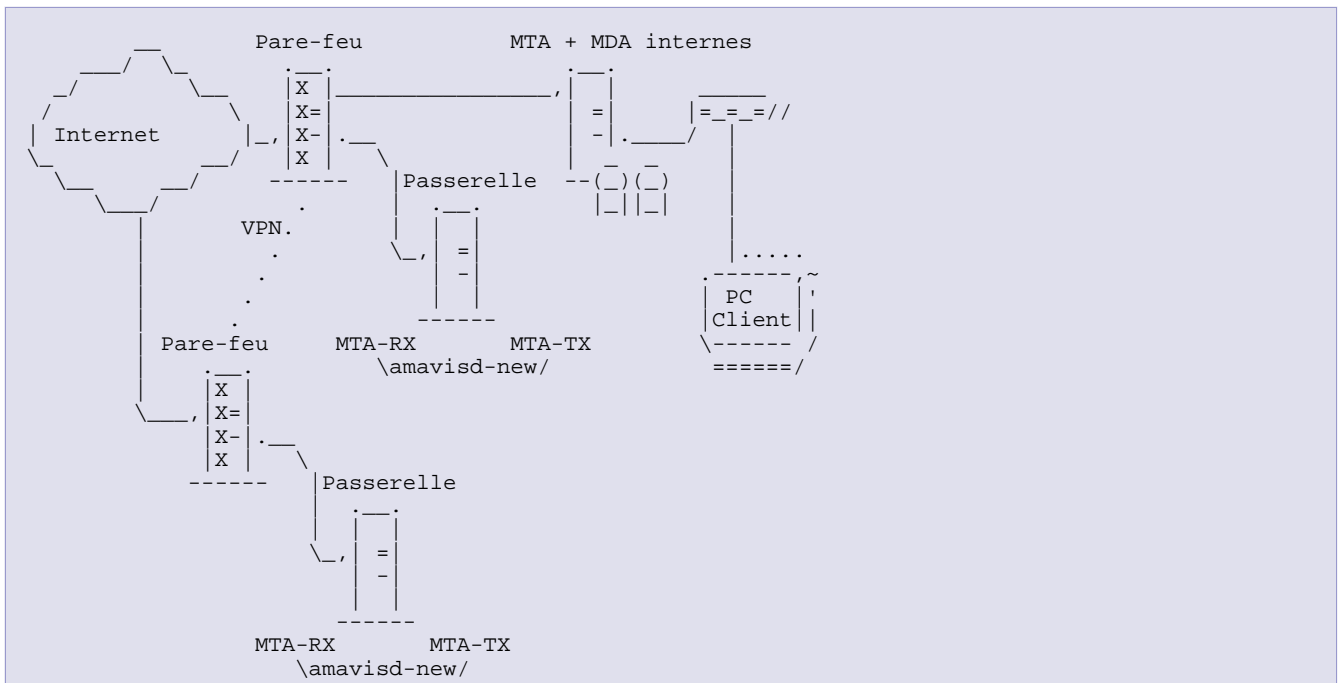
## 2.4. L'architecture réseau

Voici une architecture réseau «simple» avec laquelle l'acheminement du courrier suit les processus suivants :

1. Le *Mail Transfer Agent* de réception de la *passerelle* est désigné comme service prioritaire dans la configuration du service de noms de domaines (champ MX du fichier de configuration de zone).
2. Les règles du *Pare-feu* sont rédigées de façon à ce que les courriers électroniques issus de l'Internet arrivent à la passerelle et nulle part ailleurs.
3. Le service **amavisd-new** appelle tous les traitements voulus et prend une décision sur chaque courrier : passage au MTA suivant, mise en quarantaine ou destruction.
4. Dans le cas où le courrier doit passer au MTA suivant, les règles du *Pare-feu* sont rédigées de façon à ce que les courriers électroniques arrivent au MTA interne depuis la *passerelle* pour être délivrés dans les boîtes aux lettres des utilisateurs.



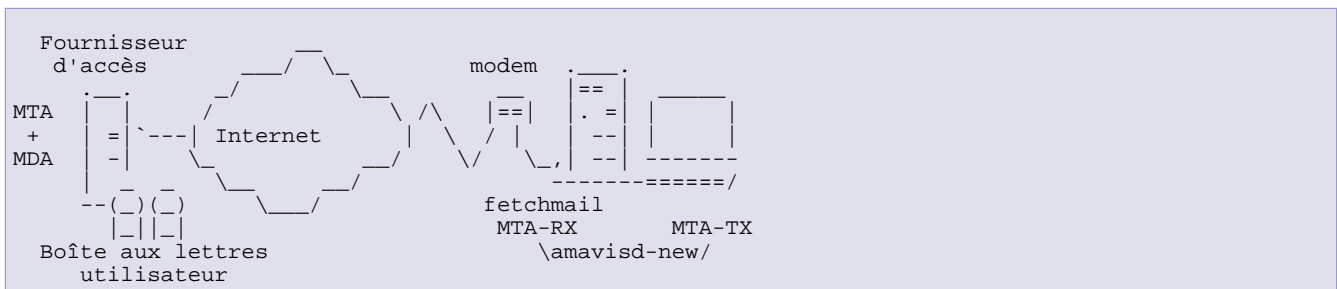
Voici une architecture réseau «plus réaliste» qui intègre une redondance de la sécurisation du service ; on parle aujourd'hui de haute disponibilité. L'acheminement du courrier doit toujours disposer de 2 voies de communication distinctes de façon à tolérer une panne quelconque sur une ou plusieurs fonctions de sécurité. On utilise alors 2 passerelles placées sur des sites géographiques différents.



Le fonctionnement des passerelles est identique à celui décrit [ci-dessus](#). Ce sont les champs MX du service de noms de domaines qui régissent les priorités entre les 2 passerelles. Un réseau privé virtuel (VPN) assure les communications entre les 2 pare-feux.

## 2.5. L'architecture «domestique»

Comment appliquer la stratégie de sécurité présentée lorsque l'on ne gère pas un domaine réseau en propre ? Il est tout à fait possible de «réintroduire» le courrier électronique déposé chez un fournisseur d'accès vers un *Mail Transfer Agent*. Voici un exemple d'architecture :



On utilise un outil particulier : **fetchmail**, dont la fonction de base est la collecte à distance et la retransmission du courrier électronique. L'outil fetchmail peut être utilisé comme passerelle POP ou IMAP pour un domaine DNS entier.

Cette architecture «domestique» peut paraître d'une complexité trop importante relativement à la charge utile de traitement du courrier électronique. Il faut cependant prendre en compte les arguments suivants :

- La première approche de sécurisation consiste à faire appels aux fonctions antispam et antivirus à partir du *Mail delivery Agent*. Comme indiqué au point **MDA(3)**, le coût d'exécution des fonctions de sécurité gêne considérablement l'utilisateur dans la consultation du courrier électronique. Cette gêne est d'autant plus importante qu'il n'existe pas de relations entre les fonctions de sécurité. Un même courrier doit passer par toutes les fonctions avant qu'une décision soit prise.
- Il ne faut pas oublier que si les «pourriels» (et/ou) les virus arrivent jusqu'à l'interface réseau de votre poste, c'est que les opérateurs n'ont pas joué leur rôle. Il n'est pas rare aujourd'hui, de devoir rejeter plus de 50% du courrier sur 2 adresses (FIXME: voir échantillon amavis-stats).

Dans ces conditions, la mise en place de cette configuration lourde à titre domestique se justifie pleinement.

## 2.6. Les règles de filtrage réseau

En considérant les 2 propositions d'architecture ci-dessus, voici un tableau récapitulatif des règles de filtrage à appliquer sur le(s) pare-feu(x).

**Tableau 1. Filtrage réseau**

Source	Destination	Type	Port	Description
Internet	Passerelle	TCP	22	Accès SSH vers la passerelle
Internet	Passerelle	TCP	25	Courriers entrant vers la passerelle
Passerelle	MTA interne et Internet	TCP	25	Courriers sortant de la passerelle
MTA interne	Passerelle	TCP	25	Courriers sortant du MTA interne
Passerelle	Internet	TCP	80	Téléchargement de fichiers Web
Passerelle	Internet	TCP	2703	Accès aux serveurs Razor
Passerelle	Internet	ICMP	8	Accès ICMP aux serveurs Razor

## 3. Le service amavisd-new

Le service `amavisd-new` n'est pas unique en son genre, mais il n'existe pas beaucoup d'équivalents. Citons juste `MIMEDefang`<sup>5</sup> qui s'appuie sur `milter`<sup>6</sup> pour composer un système de filtrage centralisé du courrier électronique.

Le service `amavisd-new` est une branche de la famille `Amavis`. Parmi les autres membres on trouve : `amavis-perl`, `amavisd` et `amavis-ng`. Aujourd'hui, le code du service `amavisd-new` est très éloigné des versions initiales d'`amavis-perl`.

Le code du service `amavisd-new` a été complètement revu. Il se distingue par l'utilisation de démons pour lesquels toutes les phases d'initialisations sont effectuées avant de commencer les traitements sur le courrier électronique. C'est ce «pré-chargement» de démon pour chaque fonction de sécurité qui assure une tenue en charge très supérieure à celle obtenue avec les outils classiques qui recourent massivement au système de fichiers.

### 3.1. L'installation

Bien qu'il existe un paquet Debian dont l'évolution est présentée à la page : *amavisd-new source package*<sup>7</sup>, celui-ci n'est pas mis à jour assez fréquemment pour pouvoir coller à l'actualité de la sécurité du courrier électronique. On touche ici à la question sensible de la *Veille Sécurité* (FIXME section journalisation). Il est possible que dans les semaines à venir la maintenance du paquet Debian rattrape son retard. Il serait alors préférable de l'utiliser. Depuis Janvier 2004, les vagues de (virus|vers) ont précipité les évolutions du service `amavisd-new`. Dans le cas de l'utilisation de `milter` (FIXME section sendmail), il est indispensable d'utiliser les sources à la place du paquet.

Le présent document décrit donc l'installation du service `amavisd-new` à partir de ses sources disponibles à la page : *amavisd-new*<sup>8</sup>. Comme la distribution utilisée est *Debian GNU/Linux*, l'organisation des fichiers et répertoires du service essaie de se conformer aux recommandations de la *Charte Debian*.

Enfin, ce qui suit ne peut se substituer à la documentation officielle sur l'installation du service : le fichier `INSTALL`<sup>9</sup>.

#### 3.1.1. Le téléchargement des sources et les dépendances

Rien de bien original pour ce qui concerne l'obtention des sources :

```
$ wget http://www.ijs.si/software/amavisd/amavisd-new-20030616-p9.tar.gz
$ su
# mv amavisd-new-20030616-p9.tar.gz /usr/local/src
# cd /usr/local/src
# mv amavisd-new-20030616 amavisd-new-20030616.old
# tar xvzf amavisd-new-20030616-p9.tar.gz
```

Pour ce qui est des autres logiciels nécessaires au fonctionnement du service, il faut comparer la liste fournie dans le fichier `INSTALL`<sup>10</sup> et les dépendances du paquet Debian. Voici ce que l'on obtient avec une installation Debian/testing :

1. On liste les dépendances de paquets :

```
$ apt-cache depends amavisd-new
amavisd-new
Dépend: adduser
Dépend: file
Dépend: libmime-perl
Dépend: libconvert-tnef-perl
Dépend: libconvert-uulib-perl
Dépend: libcompress-zlib-perl
Dépend: libarchive-tar-perl
Dépend: libarchive-zip-perl
Dépend: libmailtools-perl
Dépend: libunix-syslog-perl
Dépend: libnet-perl
Dépend: libnet-server-perl
Dépend: <libtime-hires-perl>
```

<sup>5</sup> <http://www.mimedefang.org/>

<sup>6</sup> [http://www.milter.org/milter\\_api/](http://www.milter.org/milter_api/)

<sup>7</sup> <http://packages.qa.debian.org/a/amavisd-new.html>

<sup>8</sup> <http://www.ijs.si/software/amavisd/>

<sup>9</sup> <http://www.ijs.si/software/amavisd/INSTALL>

<sup>10</sup> <http://www.ijs.si/software/amavisd/INSTALL>

```

perl
Dépend: <libdigest-md5-perl>
perl
Dépend: <libmime-base64-perl>
perl
Dépend: perl
Suggère: spamassassin
Suggère: clamav
Suggère: clamav-daemon
Suggère: lha
Suggère: arj
Suggère: unrar
Suggère: zoo
Suggère: nomarch
Suggère: cpio
Suggère: lzop
Suggère: apt-listchanges
Est en conflit avec: <amavis>
Remplace: <amavis>
amavisd-new

```

2. On affiche les versions de la liste des paquets ci-dessus pour faire la correspondance avec le fichier `INSTALL`<sup>11</sup> :

```

$ dpkg -l adduser file libconvert-tnef-perl \
libconvert-uulib-perl libcompress-zlib-perl \
libarchive-tar-perl libarchive-zip-perl \
libmailtools-perl libunix-syslog-perl libnet-perl \
libnet-server-perl perl spamassassin clamav \
clamav-daemon lha arj unrar zoo nomarch cpio \
lzop apt-listchanges
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder
| État=Non/Installé/fichier-Config/dépaqUeté/échec-conFig/H=semi
|/ Err?=(aucune)/H=à garder/besoin Réinstallation/X=les deux
||/ Nom                               Version
+++=====
ii  adduser                             3.52
ii  file                                 4.07-2
ii  libconvert-tnef-perl                 0.17-3
ii  libconvert-uulib-perl                1.0.1-1
ii  libcompress-zlib-perl                1.16-1.1
ii  libarchive-tar-perl                  1.08-1
ii  libarchive-zip-perl                  1.05-1
ii  libmailtools-perl                    1.62-1
ii  libunix-syslog-perl                  0.100-2
ii  libnet-perl                           1.18-1
ii  libnet-server-perl                   0.85-3
ii  perl                                 5.8.3-3
ii  spamassassin                          2.63-1
ii  clamav                                0.67-7
ii  clamav-daemon                         0.67-7
ii  lha                                    1.14i-2
ii  arj                                    3.10.19-2
ii  unrar                                 3.3.6-1
ii  zoo                                    2.10-9
ii  nomarch                               1.3-2
ii  cpio                                  2.5-1.1
ii  lzop                                  1.01-1
ii  apt-listchanges                       2.54

```

Ces numéros version évoluent régulièrement. Les indications de liste ci-dessus ne sont pas à prendre au pied de la lettre. On retrouve dans le journal de démarrage du service, les versions reconnues par amavisd-new (FIXME démarrage amavisd-new).

## 4. Les antivirus

À l'heure actuelle, utiliser une seule source de signature de virus n'est pas très professionnel. De nombreux exemples ont montré qu'il est très utile d'utiliser plusieurs antivirus en parallèle pour le traitement du courrier électronique.

- La réactivité des équipes techniques des fournisseurs d'antivirus doit être inférieure à la dizaine de minutes. Face à une contrainte aussi « infernale », il est tout à fait normal qu'un nouveau (ver|virus) échappe à la vigilance d'un fournisseur. Il est donc logique de chercher à mettre toutes les chances de son côté en utilisant plusieurs antivirus de (sources|marques) différentes.
- Tous les antivirus ne fonctionnent pas de la même façon. Il existe plusieurs techniques d'analyse qui donnent des résultats différents suivant les modes de propagation des codes malveillants. Là encore, il est logique de chercher à mettre toutes les chances de son côté ...

<sup>11</sup> <http://www.ijs.si/software/amavisd/INSTALL>

## 4.1. Sophos

**Sophos™** est un fournisseur d'antivirus propriétaire non libre.

### 4.1.1. SAV interface

En plus des outils antivirus propriétaires traditionnels, Sophos™ fournit un jeu de bibliothèques qui permettent de développer un produit tiers. C'est ce jeu de bibliothèques appelé *SAV interface* qui est utilisé par le démon *sophie*. Voir [Section 4.1.2, « Sophie »](#).

Chaque mois, il faut télécharger une nouvelle archive *SAV interface* pour que le démon *sophie* puisse bénéficier des nouveaux développements du fournisseur.

Téléchargement de l'archive *SAV interface*

À partir du site [Mises à jour Sophos™](#), on télécharge l'archive `linux.intel.libc6.glibc.2.2.tar.Z`.

Installation du jeu de bibliothèques

Il s'agit essentiellement de copier des fichiers dans l'arborescence `/usr/local/`.

```
phil@MailGw:~$ tar xvzf linux.intel.libc6.glibc.2.2.tar.Z
<snipped/>
phil@MailGw:~$ cd sav-install
phil@MailGw:~/sav-install$ su
Password:
MailGw:/home/phil/sav-install# ./install.sh -v
Utilitaire d'installation de Sophos Anti-Virus [[Linux/Intel]]
Copyright (c) 1998-2005 Sophos Plc, Oxford, Angleterre

Initialisation...

Les binaires seront installés dans '/usr/local/bin'
Les bibliothèques seront installées dans '/usr/local/lib'
Les pages man seront installées dans '/usr/local/man'
Les données virales seront installées dans '/usr/local/sav'
Le texte de message sera installé dans '/usr/local/sav'

SWEEP sera installé
InterCheck ne sera pas installé
<snipped/>
Ajustement en cours de /etc/sav.conf
<snipped/>
```

Purge des anciens fichiers

Les mises à jour mensuelles et le script `sophos-ide-update.pl` renomment les fichiers de signatures virales et les bibliothèques sans les effacer. Le répertoire `/usr/local/sav/` doit donc être purgé régulièrement.

Voici un exemple de script exécuté chaque mois :

```
MailGw:/etc/cron.monthly# cat sav-cleanup
#!/bin/sh

[ -d /usr/local/sav/ ] || exit 0

find /usr/local/sav/ -mtime +90 -exec rm {} \;
```

### 4.1.2. Sophie

**sophie** est un démon libre qui utilise les bibliothèques Sophos™. Associé au service **amavisd-new**, ce démon joue le rôle d'antivirus *primaire* travaillant directement en mémoire vive (RAM). Ce mode de fonctionnement allie efficacité et vitesse de traitement.

Téléchargement des sources

Le site principal **sophie** indique la version la plus récente du démon.

```
phil@MailGw:~$ wget http://www.clanfield.info/sophie/sophie-3.04.tar.bz2
phil@MailGw:~$ su
Password:
MailGw:/home/phil# mv sophie-3.04.tar.bz2 /usr/local/src
MailGw:/home/phil# cd /usr/local/src
```

```
MailGw:/usr/local/src# tar xvjf sophie-3.04.tar.bz2
```

## 5. Les échantillons relevés

Voici quelques échantillons relevés lors de l'utilisation du service **amavisd-new**. Certaines informations telles que les adresses de courrier électronique ont été masquées par des caractères x.

### 5.1. Module MIME::Tools

Ces 2 exemples sont issus de deux installations du type : sendmail - amavis-milter - amavisd-new.

Il est important de disposer d'un module de désassemblage MIME de bonne qualité pour pouvoir traiter les cas anormaux. Il faut absolument éviter qu'un message à la composition MIME volontairement erronée suffise à sauter les appels aux antivirus. Dans le cas ci-dessous, un 'W32/Dumaru-Y' est «accroché» à un message avec un en-tête MIME erroné.

```
mai 5 10:38:11 tuxbox amavis[8378]: (i458c9dV005475) warning - MIME::Parser
error: BadHead: couldn't parse header; ProblemNear:; name="accounts.zip";
Content-Transfer-Encoding: base64; Content-Disposition: attachment;
filename="myphoto.zip"; error: UnexpectedBound: part didn't end with expected
boundary [in multipart messag...

May 5 10:38:11 tuxbox sophie[831]: WARNING : Scan result =>
'/var/lib/amavis/amavis-milter-i458c9dV005475/parts/part-00004' infected with
virus 'W32/Dumaru-Y'
```

Dans ce second exemple, un autre message avec une mauvaise construction MIME contient un fichier joint exécutable. La configuration du service interdit la transmission de fichiers .exe par courrier électronique.

```
mai 5 10:29:54 tuxbox amavis[22332]: (i458Trqb012919) warning - MIME::Parser
error: UnexpectedBound: part didn't end with expected boundary [in multipart
message]; EOStoken: CLOSE 3692B18230B.1083745793/XXXXXXXXX.XXXXX.XXX; EOStype:
EXT; error: SeveredParts: unexpected end of parts before epilogue [in multipart
message]; Virtu...

mai 5 10:29:54 tuxbox amavis[22332]: (i458Trqb012919) NOTICE: DSN contains
BANNED NAME; bounce is not bouncable, mail intentionally dropped

mai 5 10:29:54 tuxbox amavis[22332]: (i458Trqb012919) BANNED name/type (.exe),
<> -> <XXXX@XXXX.XXXXXXXXXX.XX>, quarantine
virus-20040505-102954-i458Trqb012919, Message-ID:
<20040505082953.5687E182367@XXXXXXXXX.XXXXX.XXX>, Hits: -
```

## 6. Documents de référence

amavisd-new

**amavisd-new**<sup>12</sup> : site principal de l'outil de filtrage de contenu de courrier électronique. L'interface amavisd-new se place entre les gestionnaires de mise en file d'attente de réception et d'émission de l'agent de transfert e courrier électronique (MTA).

Amavis

Page principale du projet d'origine : **amavis**<sup>13</sup>. Cet outil n'est plus supporté/maintenu.

fetchmail

*The fetchmail Home Page*<sup>14</sup> : fetchmail est un utilitaire de récupération et de retransmission de courrier électronique prévu pour fonctionner avec des connexions à la demande de type PPP.

Sophos™

**Sophos™**<sup>15</sup> : fournisseur d'antivirus propriétaire non libre. Une fois la licence du logiciel acquise, Sophos™ fournit une interface d'intégration du moteur antivirus dans un logiciel tiers. Cette interface appelée SAV

<sup>12</sup> <http://www.ijs.si/software/amavisd/>

<sup>13</sup> <http://www.amavis.org/>

<sup>14</sup> <http://www.catb.org/~esr/fetchmail/>

<sup>15</sup> <http://www.sophos.fr/>

*Interface* est utilisée par **sophie** pour constituer un démon antivirus primaire pour **amavisd-new**.

Mises à jour Sophos™

<http://www.sophos.fr/support/updates><sup>16</sup> : page Web de téléchargement des mises à jour du jeu de bibliothèques Sophos™ utilisé par **sophie**.

**sophie**

**sophie**<sup>17</sup> : démon libre d'interfaçage du moteur antivirus **Sophos™**. L'objectif de ce démon est d'optimiser les appels au moteur antivirus **Sophos™**. Chaque partie d'un message est examinée à partir d'un code antivirus déjà chargé en mémoire vive (RAM).

*Debian GNU/Linux*

*Debian GNU/Linux*<sup>18</sup> : page principale de la distribution.

*Charte Debian*

*La Charte Debian*<sup>19</sup> décrit l'organisation des ressources du système d'exploitation.

---

<sup>16</sup> <http://www.sophos.fr/support/updates>

<sup>17</sup> <http://www.clanfield.info/sophie/>

<sup>18</sup> <http://www.debian.org/>

<sup>19</sup> <http://www.debian.org/doc/devel-manuals#policy>